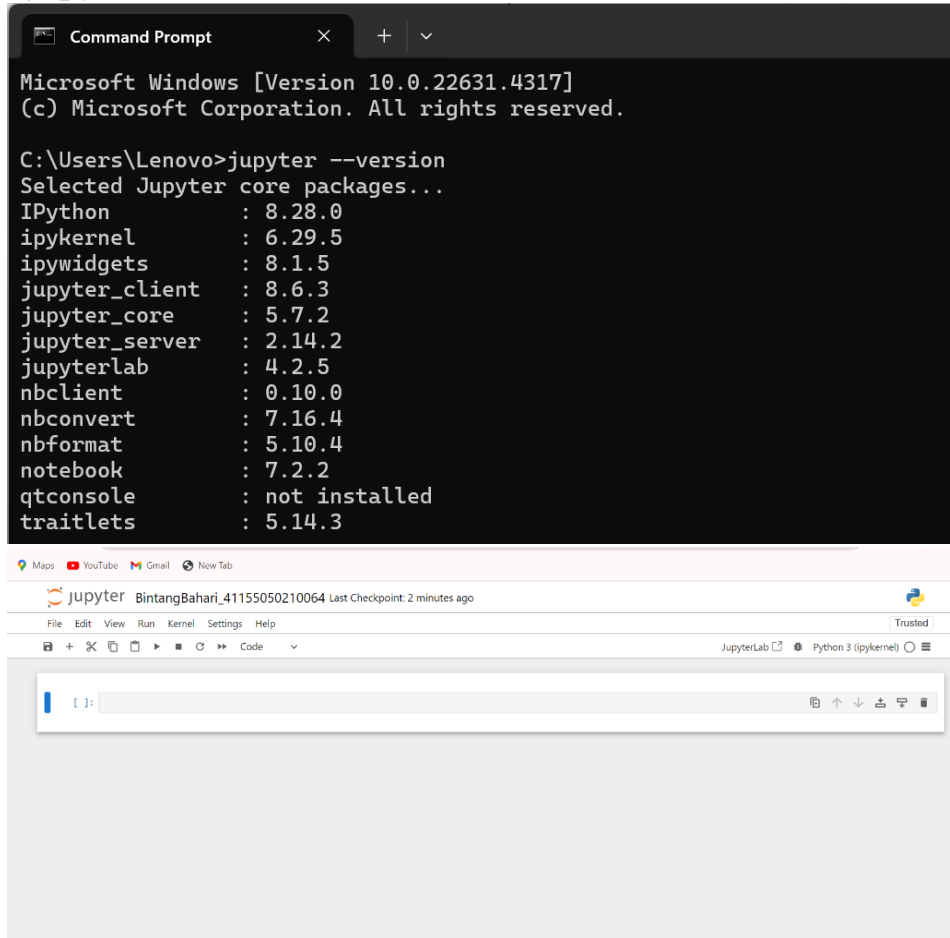


Nama : Bintang Bahari
Kelas : Informatika A2
Npm : 41155050210064

Tugas 1 Machine Learning

1. Buat screenshot sebagai jawaban nomor 1 di Tugas Pertemuan 1!
*jupyter Notebook

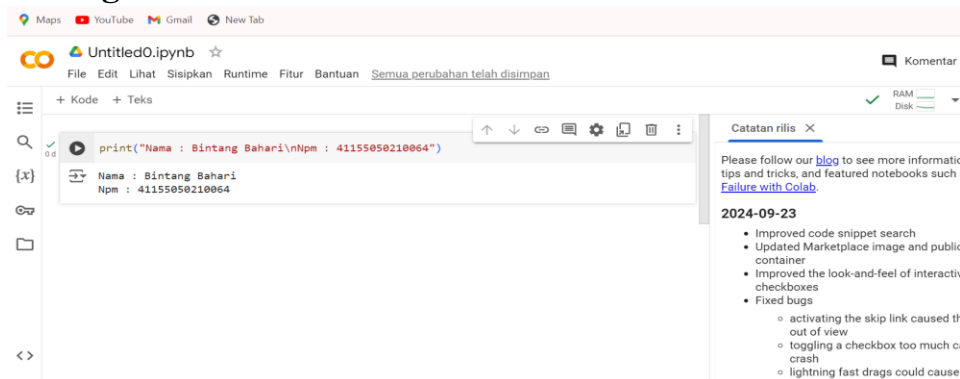


```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Lenovo>jupyter --version
Selected Jupyter core packages...
IPython          : 8.28.0
ipykernel        : 6.29.5
ipywidgets       : 8.1.5
jupyter_client   : 8.6.3
jupyter_core     : 5.7.2
jupyter_server   : 2.14.2
jupyterlab       : 4.2.5
nbclient         : 0.10.0
nbconvert        : 7.16.4
nbformat         : 5.10.4
notebook         : 7.2.2
qtconsole        : not installed
traitlets        : 5.14.3
```

Jupyter Notebook interface showing the command prompt output and the Jupyter Notebook environment.

2. Google Collab



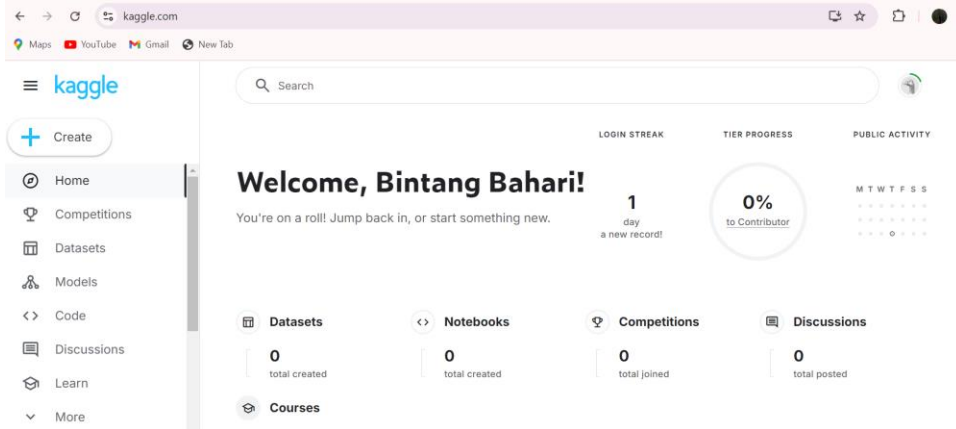
```
print("Nama : Bintang Bahari\nNpm : 41155050210064")
```

Output:

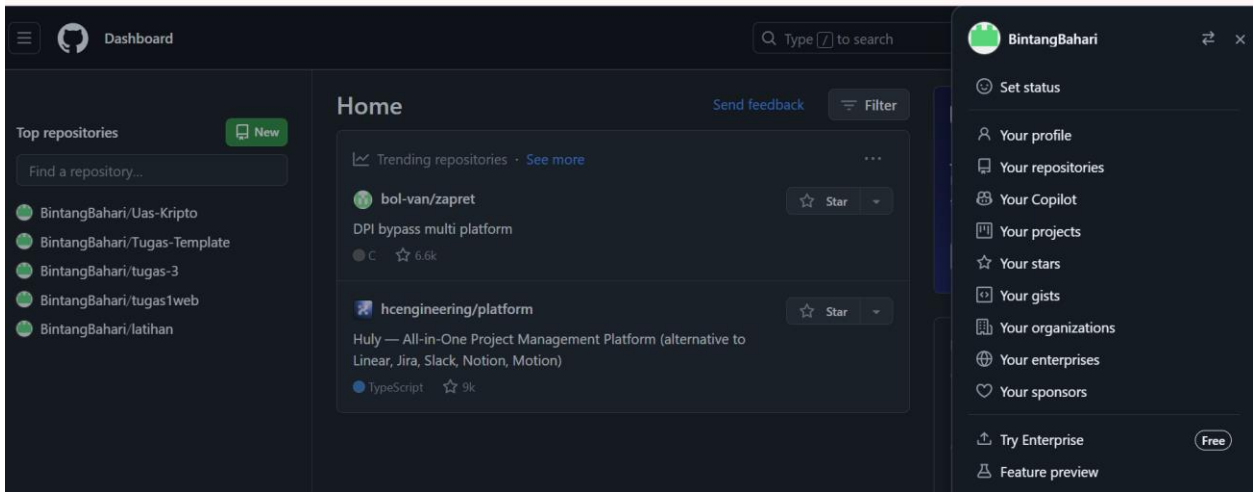
```
Nama : Bintang Bahari
Npm : 41155050210064
```

Google Colab interface showing the code editor and the console output area.

3. Buatlah akun Kaggle

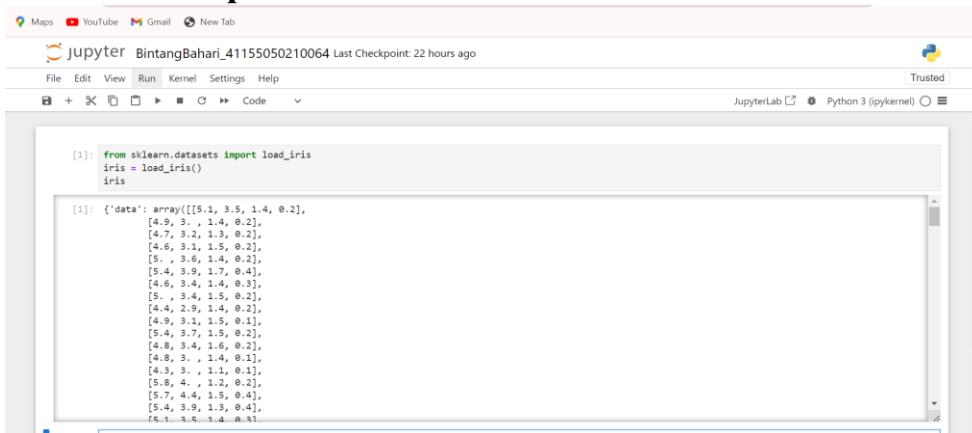


4. akun github



5.. Lakukan praktek dari <https://youtu.be/mSO2hJln0OY?feature=shared> . Praktek tersebut yaitu:

5.1. Load sample dataset



5.2. Metadata | Deskripsi dari sample dataset

```
[2]: iris.keys()

[2]: dict_keys(['data', 'target', 'frame', 'target_names', 'DESCR', 'feature_names', 'filename', 'data_module'])

[3]: print(iris.DESCR)

.. _iris_dataset:

Iris plants dataset
-----

**Data Set Characteristics:**

: Number of Instances: 150 (50 in each of three classes)
: Number of Attributes: 4 numeric, predictive attributes and the class
: Attribute Information:
  - sepal length in cm
  - sepal width in cm
  - petal length in cm
  - petal width in cm
  - class:
    - Iris-Setosa
    - Iris-Versicolour
    - Iris-Virginica

: Summary Statistics:
```

5.3. Explanatory & Response Variables | Features & Target

```
[4]: x = iris.data  
      x.shape
```

```
[4]: (150, 4)
```

```
[5]: x = iris.data  
      x
```

```
[5]: array([[5.1, 3.5, 1.4, 0.2],  
          [4.9, 3., 1.4, 0.2],  
          [4.7, 3.2, 1.3, 0.2],  
          [4.6, 3.1, 1.5, 0.2],  
          [5., 3.6, 1.4, 0.2],  
          [5.4, 3.9, 1.7, 0.4],  
          [4.6, 3.4, 1.4, 0.3],  
          [5., 3.4, 1.5, 0.2],  
          [4.4, 2.9, 1.4, 0.2],  
          [4.9, 3.1, 1.5, 0.1],  
          [5.4, 3.7, 1.5, 0.2],  
          [4.8, 3.4, 1.6, 0.2],  
          [4.8, 3., 1.4, 0.1],  
          [4.3, 3., 1.1, 0.1],  
          [5.8, 4., 1.2, 0.2],  
          [5.7, 4.4, 1.5, 0.4],  
          [5.4, 3.9, 1.3, 0.4],  
          [6.1, 3.6, 1.4, 0.3]])
```

```
[6]: y = iris.target  
      y.shape
```

```
[6]: (150,)
```

```
[7]: y = iris.target  
      y
```

```
[7]: array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
         0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

5.4. Feature & Target Names

2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2])

```
[8]: feature_names = iris.feature_names
     feature_names
```

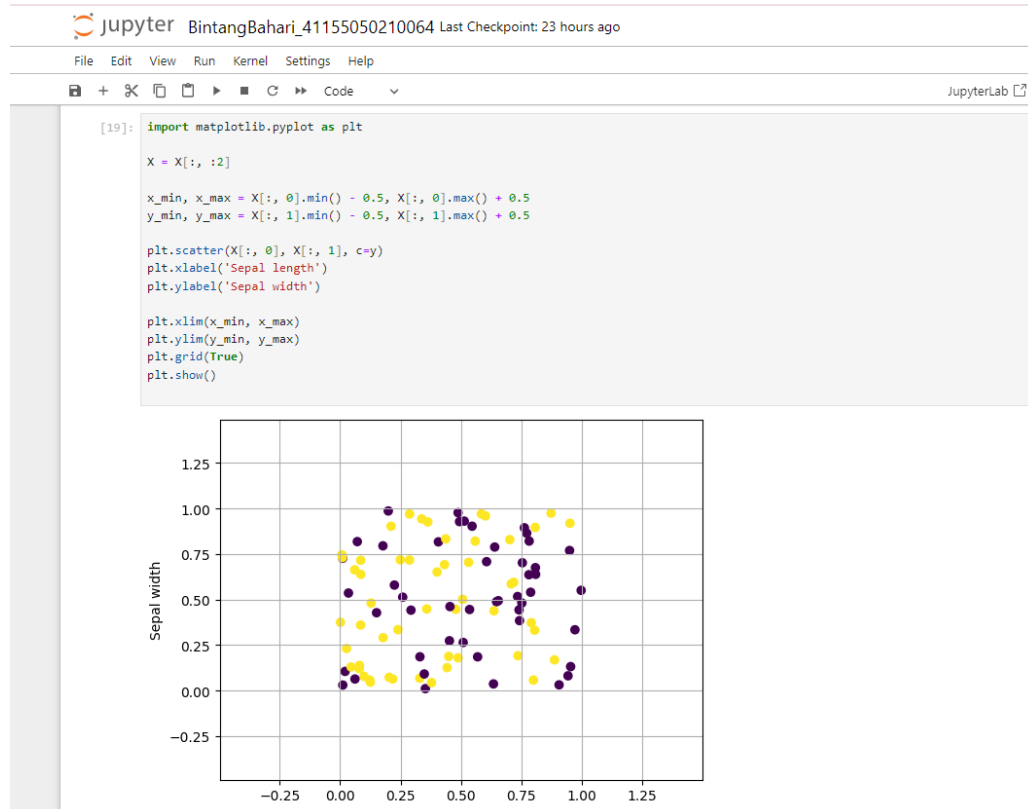
```
[8]: ['sepal length (cm)',
      'sepal width (cm)',
      'petal length (cm)',
      'petal width (cm)']
```

```
[9]: feature_names = iris.target_names
     feature_names
```

```
[9]: array(['setosa', 'versicolor', 'virginica'], dtype='<U10')
```

$$[\]: \boxed{\quad}$$

5.5. Visualisasi Data



5.6. Training Set & Testing Set

```
[24]: from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size=0.3,
                                                    random_state=1)

print(f'X train: {X_train.shape}')
print(f'X test: {X_test.shape}')
print(f'y train: {y_train.shape}')
print(f'y test: {y_test.shape}')
```

X train: (70, 2)
X test: (30, 2)
y train: (70,)
y test: (30,)

5.7. Load sample dataset sebagai Pandas Data Frame

```
[29]: iris = load_iris(as_frame=True)
```

```
iris_feature_df = iris.data  
iris_feature_df
```

```
[29]:
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2
...
145	6.7	3.0	5.2	2.3
146	6.3	2.5	5.0	1.9
147	6.5	3.0	5.2	2.0
148	6.2	3.4	5.4	2.3
149	5.9	3.0	5.1	1.8

150 rows × 4 columns

6.0. Lakukan praktek dari

<https://youtu.be/tiREcHrtDLo?feature=shared> . Praktek tersebut yaitu:

6.1. Persiapan dataset | Loading & splitting dataset

```
[6]: from sklearn.datasets import load_iris
```

```
iris = load_iris()
```

```
X = iris.data
```

```
y = iris.target
```

```
[7]: from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(X,  
                                                    y,  
                                                    test_size=0.4,  
                                                    random_state=1)
```

6.2. Training model Machine Learning

```
[12]: from sklearn.neighbors import KNeighborsClassifier
```

```
model = KNeighborsClassifier(n_neighbors=3)  
model.fit(X_train, y_train)
```

```
[12]: KNeighborsClassifier
```

```
KNeighborsClassifier(n_neighbors=3)
```

6.3. Evaluasi model Machine Learning

```
[14]: from sklearn.metrics import accuracy_score
```

```
y_pred = model.predict(X_test)
acc = accuracy_score(y_test, y_pred)
print(f'Accuracy: {acc}')
```

```
Accuracy: 0.9833333333333333
```

6.4. Pemanfaatan trained model machine learning

```
[15]: data_baru = [[4, 6, 2, 2],
                  [2, 3, 3, 5]]
```

```
preds = model.predict(data_baru)
preds
```

```
[15]: array([0, 1])
```

```
[16]: pred_species = [str(iris.target_names[p]) for p in preds]
print(f'Hasil Prediksi: {pred_species}')
```

```
Hasil Prediksi: ['setosa', 'versicolor']
```

```
[ ]: |
```

6.5. Deploy model Machine Learning | Dumping dan Loading model

```
[18]: import joblib
```

```
joblib.dump(model, 'iris_classifier_knn.joblib')
```

```
[18]: ['iris_classifier_knn.joblib']
```

```
[19]: production_model = joblib.load('iris_classifier_knn.joblib')
```

```
[ ]: |
```

7.0. Lakukan praktek dari

<https://youtu.be/smNnhEd26Ek?feature=shared> . Praktek tersebut yaitu:

7.1. Persiapan sample dataset

```
[2]: import numpy as np
      from sklearn import preprocessing

      sample_data = np.array([[2.1, -1.9, 5.5],
                              [-1.5, 2.4, 3.5],
                              [0.5, -7.9, 5.6],
                              [5.9, 2.3, -5.8]])

      sample_data
```

```
[2]: array([[ 2.1, -1.9,  5.5],
            [-1.5,  2.4,  3.5],
            [ 0.5, -7.9,  5.6],
            [ 5.9,  2.3, -5.8]])
```

```
[3]: sample_data.shape
```

```
[3]: (4, 3)
```

7.2. Teknik data preprocessing 1: binarisation

```
[4]: preprocessor = preprocessing.Binarizer(threshold=0.5)
      binarised_data = preprocessor.transform(sample_data)
      binarised_data
```

```
[4]: array([[1., 0., 1.],
            [0., 1., 1.],
            [0., 0., 1.],
            [1., 1., 0.]])
```

7.3. Teknik data preprocessing 2: scaling

```
[5]: sample_data

[5]: array([[ 2.1, -1.9,  5.5],
            [-1.5,  2.4,  3.5],
            [ 0.5, -7.9,  5.6],
            [ 5.9,  2.3, -5.8]])

[8]: preprocessor = preprocessing.MinMaxScaler(feature_range=(0,1))
preprocessor.fit(sample_data)
scaled_data = preprocessor.transform(sample_data)
scaled_data

[8]: array([[0.48648649, 0.58252427, 0.99122807],
            [0.          , 1.          , 0.81578947],
            [0.27027027, 0.          , 1.          ],
            [1.          , 0.99029126, 0.          ]])

[9]: scaled_data = preprocessor.fit_transform(sample_data)
scaled_data

[9]: array([[0.48648649, 0.58252427, 0.99122807],
            [0.          , 1.          , 0.81578947],
            [0.27027027, 0.          , 1.          ],
            [1.          , 0.99029126, 0.          ]])
```

```
1: |
```

7.4. Teknik data preprocessing 3: normalisation

```
[10]: sample_data

[10]: array([[ 2.1, -1.9,  5.5],
            [-1.5,  2.4,  3.5],
            [ 0.5, -7.9,  5.6],
            [ 5.9,  2.3, -5.8]])

[12]: li_normalised_data = preprocessing.normalize(sample_data, norm='l1')
li_normalised_data

[12]: array([[ 0.22105263, -0.2          ,  0.57894737],
            [-0.2027027 ,  0.32432432,  0.47297297],
            [ 0.03571429, -0.56428571,  0.4          ],
            [ 0.42142857,  0.16428571, -0.41428571]])

[13]: l2_normalised_data = preprocessing.normalize(sample_data, norm='l2')
l2_normalised_data

[13]: array([[ 0.33946114, -0.30713151,  0.88906489],
            [-0.33325106,  0.53320169,  0.7775858 ],
            [ 0.05156558, -0.81473612,  0.57753446],
            [ 0.68706914,  0.26784051, -0.6754239 ]])
```

```
[ ]: |
```